<u>IN THE CLAIMS</u>:

1. (Previously Presented) A method of converting code to a hardware realization, the method comprising steps of:

receiving user code including at least one algorithm specification, at least one data representation specification, and at least one data communication specification;

compiling the user code, wherein the user code is used to create a digital circuit; and

wherein the at least one algorithm specification, the at least one data representation specification, and the at least one data communication specification are independent and each of these specifications are modifiable without affecting the others.

2. (Original) The method of claim 1, wherein the step of compiling further comprises steps of:

compiling the user code to generate a netlist; and

mapping the netlist using physical design tools for creating the digital circuit.

3. (Original) The method of claim 1, further comprising a step of creating the digital circuit based on the user code.

4. (Original) The method of claim 3, wherein the step of creating further comprises steps of:

creating configuration data; and

subsequently configuring an FPGA.

5. (Original) The method of claim 3, wherein the step of creating further comprises creating a specification for one of a custom-designed VLSI chip and a standard cell VLSI chip.

6. (Original) The method of claim 1, wherein the step of compiling further comprises retrieving information from libraries, the information being associated with the at least one algorithm specification, the at least one data representation specification, and the at least one data communication specification.

7. (Original) The method of claim 1, wherein the at least one algorithm specification includes at least one variable having a defined set of values not varying by platform and at least one operator having a function not varying by platform.

8. (Original) The method of claim 1, wherein the at least one data representation specification includes one of 2's-complement, signed-digit, and fully-redundant carry-save for each variable in the at least one algorithm specification.

9. (Original) The method of claim 1, wherein the at least one data communication specification includes one of bit-serial, digit-serial and fully-parallel for each variable in the at least one algorithm specification.

10. (Original) The method of claim 1, wherein the step of receiving the user code further comprising receiving selections of the at least one algorithm specification, the at least one data representation specification, and the at least one data communication specification from a graphical user interface (GUI).

11. (Canceled).

12. (Previously Presented) A method of converting code to a hardware realization, the method comprising steps of:

receiving user code, wherein the user code is an expression of a design of a digital circuit;

identifying variables used in an operation in the user code, the operation including at least one operator;

identifying a set of assumable values for each of the identified variables;

calculating a set of assumable values for other variables in the user code holding the results of the operation based on the identified set of assumable values; and

compiling the user code.

13. (Original) The method of claim 12, wherein the user code further includes at least one algorithm specification, at least one data representation specification, and at least one data communication specification.

14. (Original) The method of claim 13, wherein the at least one data representation specification includes one of 2's-complement, signed-digit, and fully-redundant carry-save for each variable in the at least one algorithm specification.

15. (Original) The method of claim 13, wherein the at least one algorithm specification, the at least one data representation specification, and the at least one data communication specification are independent, such that each of these specifications are modifiable without affecting the others.

16. (Previously Presented) A system operable to create a digital circuit from user code, the system comprises:

a compiler compiling the user code, the user code including at least one algorithm specification, at least one data representation specification, and at least one data communication specification; wherein

the at least one algorithm specification, the at least one data representation specification, and the at least one data communication specification are independent, such that each of these specifications are modifiable without affecting the others.

17. (Original) The system of claim 16, further comprising a plurality of libraries connected to the compiler, the plurality of libraries including information associated with the at least one algorithm specification, the at least one data representation specification, and the at least one data communication specification, wherein the compiler retrieves the information to compile the user code.

18. (Original) The system of claim 16, wherein the compiler compiles the user code to generate a netlist that lists components in the digital circuit.

19. (Original) The system of claim 18, further comprising physical design tools, the physical design tools being used to create the digital circuit from the netlist.

20. (Original) The system of claim 16, wherein the at least one algorithm specification includes variables and an operator in an operation in the user code, and each of the variables includes an identifiable set of assumable values to be used in the operation.

21. (Original) The system of claim 20, wherein the compiler is operable to calculate a set of assumable values for a variable in the operation performed by an operator holding a result of the operation based on the identified set of assumable values for the variables used in the operation.

22. (Previously Presented) The method of claim 12, wherein the user code is programmed in a high level programming language.

23. (Previously Presented) The method of claim 22, wherein the high level programming language comprises one of a C-like and JAVA-like programming language.

24. (Currently Amended) A method of using software code to design a digital circuit, the method comprising:

receiving high level programming language code representing an expression of a design of a digital circuit, wherein the code comprises:

at least one algorithm specification including at least one variable and at least one operator;

at least one data representation specification identifying a type of data representation for the at least one variable in the digital circuit; and at least one data communication specification identifying a data communication implementation in the digital circuit associated with the at least one variable;

identifying variables used in an operation in the user code, the operation including the at least one operator;

identifying a set of assumable values for each of the identified variables;

calculating a set of assumable values for at least one other variable in the code holding a result of the operation based on the identified set of assumable values; and

compiling the code, the compiled code being used to create the digital circuit.

25. (Previously Presented) The method of claim 24, wherein the high level programming language comprises one of a C-like and JAVA-like programming language.

26. (Previously Presented) The method of claim 24, wherein the at least one variable has a set of assumable values defined in the code.

27. (Canceled).

28. (Previously Presented) The method of claim 24, wherein the at least one algorithm specification, the at least one data representation specification, and the at least one data communication specification are independent, such that each of these specifications are modifiable without affecting the others.

29. (Previously Presented) The method of claim 24, wherein the code identifies a defined set of values for the at least one variable that do not vary by platform.

30. (Previously Presented) Computer software embedded on a computer readable medium, the computer software comprising instructions performing:

receiving user code including at least one algorithm specification, at least one data representation specification, and at least one data communication specification, wherein the at least one algorithm specification, the at least one data representation specification, and the at least one data communication specification are independent and each of these specifications are modifiable without affecting the others; and

compiling the user code, wherein the user code is used to create a digital circuit.

31. (Previously Presented) The computer software of claim 30, further comprising instructions performing:

identifying variables used in an operation in the user code, the operation including at least one operator;

identifying a set of assumable values for each of the identified variables; and

calculating a set of assumable values for other variables in the user code holding the results of the operation based on the identified set of assumable values.

32. (Previously Presented) The computer software of claim 30, wherein the user code is programmed in a high level programming language.

33. (Previously Presented) An apparatus comprising:

means for receiving user code including at least one algorithm specification, at least one data representation specification, and at least one data communication specification, wherein the at least one algorithm specification, the at least one data representation specification, and the at least one data communication specification are independent and each of these specifications are modifiable without affecting the others; and

means for compiling the user code, wherein the user code is used to create a digital circuit.

34. (Previously Presented) The apparatus of claim 33, further comprising:

means for identifying variables used in an operation in the user code, the operation including at least one operator;

means for identifying a set of assumable values for each of the identified variables; and

means for calculating a set of assumable values for other variables in the user code holding the results of the operation based on the identified set of assumable values.

35. (Previously Presented) The apparatus of claim 33, wherein the user code is programmed in a high level programming language.